

Jiajun Sun

Service-Constraint Based Truthful Incentive Mechanisms for Crowd Sensing

Received: 2014-04-05

Abstract Crowd sensing is a new paradigm which leverages the pervasive smartphones to efficiently collect and upload sensing data, enabling numerous novel applications. To achieve good service quality for a crowd sensing application, incentive mechanisms are necessary for attracting more user participation. Most of existing mechanisms apply only for the budget-constraint scenario where the platform (the crowd sensing organizer) has a budget limit. On the contrary, we focus on a different scenario where the platform has a service limit. Based on the offline and online auction model, we consider a general problem: users submit their private profiles to the platform, and the platform aims at selecting a subset of users before a specified deadline for minimizing the total payment while a specific service can be completed. Specially, we design offline and online service-constraint incentive mechanisms for the case where the value function of selected users is monotone submodular. The mechanisms are individual rationality, task feasibility, computational efficiency, truthfulness, consumer sovereignty, constant frugality, and also performs well in practice. Finally, we use extensive simulations to demonstrate the theoretical properties of our mechanisms.

Keywords Crowd sensing · Service constraint · Incentive mechanisms · Online auction

1 Introduction

Crowd sensing is a new paradigm, which utilizes pervasive smartphones to efficiently collect and upload data. Nowadays, the proliferation of smartphones makes it possible to provide a new opportunity for extending from the virtual space (online social networks) to a larger real

physical world (Internet of Things), making users' contributions easier and omnipresent, such as Nericell [13], SignalGruru [10], and VTrack [19] for providing omnipresent traffic information, Ear-Phone [14] and NoiseTube [11] for making noise maps.

While participating in these applications, smartphone users consume their own resources such as battery and computing power, and disclose their locations with potential privacy threats. Thus, incentive mechanisms are necessary to provide participants with enough rewards for their participation costs. There are several incentive mechanism studies for guaranteeing adequate user participation in past literature. Generally, two scenarios for these incentive mechanisms were considered: the offline scenarios and online scenarios. For example, for the offline scenarios, the authors of [21] designed truthful incentive mechanisms for the user-centric model and platform-centric model respectively. For the online scenarios, the authors of [1; 17; 18] designed incentive mechanisms based on the bidding model and the posted price model for the additive utility function and submodular utility function respectively. However, all these works only applied for the scenario with the budget constraint where the platform with a fixed budget aims at maximizing the platform's utility (e.g., the total value of the tasks completed by selected users).

However, when the platform has a service limit instead of a budget limit, which indicates that the platform need to minimize the total payment for completing the fixed service, these truthful incentive mechanisms become infeasible. To address this problem, the authors of [8; 17] investigate the frugality of incentive mechanisms for the offline scenario, in which all of participating users report their profiles, including the tasks they can complete and the bids, to the platform in advance, and then the platform selects a subset of users after collecting the sensing profiles of all users to minimize its total payments under the condition that the specific tasks can be completed. But these mechanisms only apply for the linear value function of sensing tasks.

J. Sun
Beijing Key Lab of Intelligent Telecommunication Software and Multimedia, Beijing University of Posts and Telecommunications.
E-mail: jiajunsun.bupt@gmail.com, mhd@bupt.edu.cn

In this paper, we concern a more general case, where the value function of selected users' services is monotone submodular for service constraints, instead of additive function supporting the homogeneous and heterogeneous tasks. We investigate the offline and online scenarios respectively for monotone submodular for service constraints. For the offline scenario, the platform procures a optimal solution to a given sensing services while minimizing the total payment at the end of a specified deadline. For online scenario, where users always arrive in a sequential order, and user availability changes over time, so as to apply to most of the above crowd sensing applications, the platform online determines whether to select a user for a given sensing services while minimizing the total payment. For the two scenarios, we consider users who are game-theoretic and seek to make strategy (possible report a false cost or arrival/departure time) to maximize their individual utility in equilibrium. Thus, the problem of selecting feasible users while minimizing the total payment can be modeled as the offline and online auctions under the service and time constraints.

For the offline scenario, we adopt a "myopic" way to select the optimal users to minimize the total payment. As long as the utility function satisfies the submodularity, a natural diminishing returns condition, the mechanism satisfy the following critical properties: 1) Computational Efficiency: the auction can determine the winners and payments in polynomial time; 2) Individual Rationality: each user can expect a non-negative utility by participating in the auction; 3) Service Constraint: It ensures the the platform's service constraint is not violated. In this paper, service constraint requires the mechanism to satisfy: $V(S) = R$; 4) Truthfulness: no mobile user can benefit from cheating about its true valuation on its cost of participation. For the online scenario, we apply a multiple-stage sampling-accepting process to solicit bids from users. At every stage the mechanism allocates sensing tasks to an arriving smartphone user only if his marginal utility is not less than a certain threshold density that has been computed using previous users' bids and profiles as the sample set until the service is completed. The threshold density is calculated in a manner that guarantees the above desirable performance properties of the offline mechanism. Besides, the online mechanism also satisfies Constant Frugality: The mechanisms have constant frugality ratio, i.e., if it announces the fixed services with the value R in expectation while guaranteeing that the total payment is no more than the minimum cost required to achieve γR services in the offline scenario. The main contributions of this paper are summarized as follows:

- We design a service-constraint offline and online incentive mechanisms to ensure the minimal payment of the platform for performing the required services respectively.
- We rigorously prove that these incentive mechanisms are satisfying the above desirable performances. We

also evaluate the performance and validate their theoretical properties via extensive simulations.

The rest of the paper is organized as follows. In Section 2, we briefly discuss the related work and motivation. In Section 3, we present our system model and our design goals. In Section 4 and Section 5, we design two service-constraint based incentive mechanisms for the offline and online scenario respectively, followed by the performance evaluation in 6. Finally, Section 7 concludes remarks.

2 Background and Related Work

There are growing interest in investigating the incentives for users in online crowd sensing applications. For examples, the authors of [12; 15] study other, non-monetary incentives that could improve the quality of users' performance. The authors of [4] apply no regret learning to better understand users' behavior and improve the results of sensing information aggregation from crowds. In contrast, the authors of [20] study the money incentives to maximize tasks by using bandit algorithms. While it is a natural approach, they leave room for frameworks that allow better theoretical guarantees as used in this paper. The authors of [7] study an orthogonal problem and present an algorithmic framework for matching users with requesters based on their skills in crowd sensing applications.

Based on these frameworks, there are two classes of different model studied extensively. One is to design the budget-constraint truthful incentive mechanisms for stimulating adequate users to participate in crowd sensing applications. For example, the authors of [1; 17; 18; 21] designed truthful incentive mechanisms for the offline and online scenarios for maximizing the platform's utility. But these works fail to handle the incentive problem of extensive user participation under the service constraint. The other is to design service-constraint incentive mechanisms for soliciting users' true costs. For example, the authors of [8; 17] investigates the frugality of incentive mechanisms for the offline scenario with the homogeneous and heterogeneous tasks. But they do not propose feasible truthful incentive mechanisms for minimizing the total payment. On the contrary, in this paper, we are interested in studying minimizing payment online incentive mechanisms under given service constraint for the offline and online scenarios, where the value function of selected users' services is monotone submodular for service constraints.

3 System Model and Problem Formulation

3.1 System Model

We focus on crowd sensing applications with the goal to monitor some spatial phenomenon, such as air quality or traffic. We consider the following crowd sensing system model illustrated in Fig. 1. The system consists of a crowd sensing application platform, which resides in the cloud and consists of multiple sensing servers, and many mobile device users, which are connected to the cloud by cellular networks (e.g., GSM/3G/4G) or WiFi connections. The platform first publicizes a crowd sensing campaign in an area of interest (AoI), aiming at finding some users to complete a required utility value R reflecting service quality (given announced services). Then a set of users $\mathcal{U} = \{1, 2, \dots, n\}$ interested in the campaign report their profiles to the platform. Finally, the platform selects a feasible subset of users to complete the given service before the deadline T .

The platform is only interested in minimizing the total payment to the selected users under the given service limit. We denote the total services of the campaign as a finite set of locations, $\Gamma = \{\tau_1, \tau_2, \dots, \tau_m\}$, where each $\tau_i \in \Gamma$ could, e.g., denote a zip code or more fine grained street address, depending on the crowd sensing application. Each user can sense a subset Γ_i of Γ ($\Gamma_i \subseteq \Gamma$) like the number of locations depending on her geolocation or mobility as well as the type of device used, and have the cost c_i corresponding to Γ_i . All these information form the profile of user i , i.e., $\mathcal{P}_i = (c_i, \Gamma_i)$. Since smartphones are owned by different users, it is reasonable to assume that users are selfish but rational. Hence each user only wants to maximize its own utility, and will not participate in the campaign unless there is sufficient incentive.

In this paper, we study two scenarios: the offline scenario and online scenario, where the value function of selected users' services is monotone submodular for service constraints. In the offline scenario, all of participating users report their profiles to the platform synchronously, and then the platform allocates services to a subset of users by considering the profiles of all users at once. Different from the batched and synchronized manner in the offline scenario, the interactive process in the online scenario is sequential and asynchronous. Each user arrives in a sequential order and submits its profile. Receiving the profile, the platform must make an irrevocable decision about how much payment to pay to each arrival user before the user departs until reaching the service quality required. We assume that in each time step, a single user appears and the platform makes a decision that is based on the information it has about the user and the history of the previous $i - 1$ stages. Generally, there are three classes of user models: the *i.i.d. model*, the *secretary model*, and the *adversarial model*. The first model means that at each time step the costs and values of users are drawn from some unknown distributions. The second model means that the users' costs are chosen by an adversary, however their arrival order is a permu-

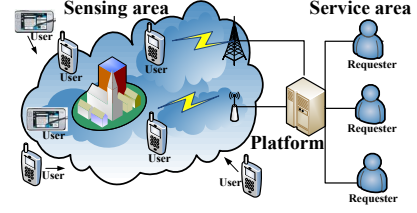


Fig. 1 Our crowd sensing system framework.

tation that is drawn uniformly at random from the set of all possible permutations. In the third model, the users' costs and their arrival order are chosen by an adversary. Note that in the third model, although the adversary cannot observe the actions the mechanism takes, since it has full knowledge, the adversary chooses the worst arrival order and costs. Thereby, the mechanism cannot obtain the optimal solutions. Thus, in this paper, we only account for the two models with respect to the distribution of users, described in increasing order of generality: the i.i.d. model and the secretary model.

3.2 Problem Formulation

We model the above service-constraint based interactive process between the platform and users as an auction with service and time constraints. Receiving the crowd sensing campaign from the platform, each user i provides its profile $\mathcal{P}_i = (c_i, \Gamma_i)$ to the platform so as to expect a payment in return for its service. Since we assume that users are game-theoretic and seek to make strategy to maximize their individual utility in equilibrium. Note that in its profile, only its service Γ_i is true so that the platform can identify whether the given services are fulfilled. That is, user i can misreport his cost, since his cost is private and only known to himself. Thus, our strategy space can allow user i to declare $\hat{\mathcal{P}}_i = (b_i, \Gamma_i)$, where b_i is a reserve price or a bid made by user i so as to sell its service. Assume that the platform has given announced services denoted as a utility value U_0 that it is willing to achieve. In order to complete the required sensing services, more formally, an offline/online mechanism $\mathcal{M} = (f, p)$, which consists of an allocation function $f : \mathcal{R}_+^n \rightarrow 2^{[n]}$ and a payment function $p : \mathcal{R}_+^n \rightarrow \mathcal{R}_+^n$, is needed. That is, for users' $\hat{\mathcal{P}} = (\hat{\mathcal{P}}_1, \hat{\mathcal{P}}_2, \dots, \hat{\mathcal{P}}_n)$, the allocation function computes an allocation of services for a feasible subset of users $\mathcal{S} \subseteq \mathcal{U}$ and the payment function returns a payment vector to feasible users. Thus, the utility of user i is $p_i - c_i$ if it is selected, 0 otherwise. The platform expect to minimize the payments while achieving the quality of announced services, i.e.,

$$\min \sum_{i \in \mathcal{S}} p_i \quad \text{Subject to } V(\mathcal{S}) \geq R$$

where $V(\mathcal{S})$ is the monotone submodular value function of services from the selected users \mathcal{S} , illustrated in the following definition.

Definition 1 (Submodular Function) Let \mathbb{N} be a finite set, a function $V : 2^\Omega \rightarrow \mathcal{R}$ is submodular if $V(S \cup \{i\}) - V(S) \geq V(T \cup \{i\}) - V(T), \forall S \subseteq T \subseteq \Omega$, where \mathcal{R} is the set of reals.

4 Offline Mechanism under the Service Constraint

In this section, we present an offline mechanism under the service constraint, satisfying the previous desirable properties.

For crowd sensing applications in the offline scenario, the authors of [16; 20; 21] apply the proportional share allocation rule proposed in [16] to address the extensive user participation issue. However, the mechanism only applies for the offline scenario with the budget constraint. To address this problem, we present a service-constraint offline incentive mechanism that satisfies the previous desirable properties. Illustrated in Algorithm 1, our mechanism consists of two phases: the winner selection phase and the payment determination phase.

Algorithm 1 OMS// An Offline Mechanism for the Service constraint

Input: User set \mathcal{U} , the service constraint R .

Output: The set of winners S .

```
// Phase 1: Winner selection under services  $R$ 
1:  $S \leftarrow \emptyset; i \leftarrow \arg \max_{j \in \mathcal{U}} V_j(S)/b_j;$ 
2: while  $V(S) < R$  do
3:    $S \leftarrow S \cup i;$ 
4:    $i \leftarrow \arg \max_{j \in \mathcal{U} \setminus S} (V_j(S)/b_j);$ 
5: end while
6:  $B \leftarrow \sum_{j \in S} b_j;$ 
// Phase 2: Winner selection under budget  $B$ 
7:  $S \leftarrow \emptyset; i \leftarrow \arg \max_{j \in \mathcal{U}} V_j(S)/b_j;$ 
8: while  $V_i(S)/b_i \geq V(S \cup i)/B$  do
9:    $S \leftarrow S \cup i;$ 
10:   $i \leftarrow \arg \max_{j \in \mathcal{U} \setminus S} (V_j(S)/b_j);$ 
11: end while
// Phase 3: Payment determination
12: for each user  $i \in \mathcal{U}$  do
13:    $p_i \leftarrow 0;$ 
14: end for
15: for each user  $i \in S$  do
16:    $\mathcal{U}' \leftarrow \mathcal{U} \setminus \{i\}; \mathcal{T} \leftarrow \emptyset;$ 
17:   repeat
18:     $i_j \leftarrow \arg \max_{j \in \mathcal{U}' \setminus \mathcal{T}} (V_j(\mathcal{T})/b_j);$ 
19:     $p_i \leftarrow \max\{p_i, \min\{b_{i(j)}, \eta_{i(j)}\}\};$ 
20:     $\mathcal{T}_{j-1} \leftarrow \mathcal{T}; \mathcal{T} \leftarrow \mathcal{T} \cup \{i_j\};$ 
21:  until  $V(\mathcal{T}) \geq R$ 
22: end for
23: return  $(S, p);$ 
```

From Definition 1, we can know the utility function V is submodular and derive the following sorting according to increasing marginal contributions relative to their bids

from users' set to find the largest k satisfying $V(S \cup k) < R$.

$$V_1/b_1 \geq V_2/b_2 \geq \dots \geq V_{|\mathcal{U}|}/b_{|\mathcal{U}|}, \quad (1)$$

where V_k denotes $V_{k|S_{k-1}} (= V(S_{k-1} \cup \{k\}) - V(S_{k-1}))$, $S_k = \{1, 2, \dots, k\}$, and $S_0 = \emptyset$. To calculate the payment of each user, we sort the users in $\mathcal{U} \setminus \{i\}$ similarly as follows:

$$V_{i_1}(\mathcal{T}_0)/b_{i_1} \geq V_{i_2}(\mathcal{T}_1)/b_{i_2} \geq \dots \geq V_{i_{n-1}}(\mathcal{T}_{n-2})/b_{i_{n-1}}, \quad (2)$$

The marginal value of user i at the position j is $BV_{i(j)}(\mathcal{T}_{j-1})/V(\mathcal{T}_j)$, where $B = \sum_{j \in S} b_j$. Assume that k' to be the position of the last user $i_j \in \mathcal{U} \setminus \{i\}$, such that $V(\mathcal{T}_{j'}) < R$. To guarantee the truthfulness, each winner should be given the payment of the critical value. This indicates that user i can not win the auction if it reports higher than this critical value. More details are given in Algorithm 1, where $b_{i(j)} = V_{i(j)}(\mathcal{T}_{j-1})b_{i_j}/V_{i_j}(\mathcal{T}_{j-1})$ and $\eta_{i(j)} = V_{i(j)}(\mathcal{T}_{j-1})B/V(\mathcal{T}_{j-1} \cup \{i\})$.

Since the OMS mechanism is very similar with MSensing in [16; 21], only with three differences. The one is that the services allocated to the winners is a constraint instead of a factor in the objective function. The second one is that OMS is a frugal mechanism instead of a budget constraint mechanism, hence introducing line 6 of Algorithm 1. But these lines' introduction has no impact on the following desirable properties. Thus, putting these together, we have the following theorem.

Theorem 1 *The OMS mechanism satisfies individual rationality, computational efficiency, service feasibility, and truthfulness under the offline scenario.*

5 Online Mechanism under the Service Constraint

In this section, we present an online mechanism for the service-constraint online scenario, satisfying all desirable properties. To facilitate understanding, it is also assumed that users arrive in a sequential order. But our mechanism can easily apply generally or be extended to an random online scenario.

5.1 Service-Constraint Online Mechanism Design

An online mechanism needs to overcome several nontrivial challenges. First, the users' costs are unknown and need to be elicited in a truthful reporting manner. Second, an announced services should be completed before the deadline. Finally, the mechanism needs to tackle the

online arrival of the users. To achieve good frugality, previous online solutions and generalized secretary problems [6; 3; 17; 9] is via sampling: the first batch of the input is rejected and used as a sample which enables making an informed decision on the rest of the users. Since users are likely to be discouraged to sense data knowing the pricing mechanism will automatically reject their bid. In other words, those users arriving early have no incentive to report their bids to the platform, which may delay the users' completion or even lead to task starvation, i.e., the consumer sovereignty issue in economics. Although the author of [9] adopts a multi-stage sampling-accepting process, it applies Dynkin's algorithm [5] for the classic secretary problem at the initial stage. Obviously, this solution also cannot ensure the above task-starvation issue, since Dynkin's algorithm adopts a two-stage sampling-accepting process.

To address the above challenges, we introduce a multi-stage sampling-accepting process to design our online incentive mechanism. At each stage, based on the above submodularity, the mechanism maintains a density threshold which is used to decide whether to accept the users' bids. The mechanism dynamically increases the sample size and learns a budget that are enough to allocate users for fulfilling the required services, then apply this budget to compute a density threshold by applying budget feasible mechanisms, and finally apply this density threshold for making further decisions.

Specifically, our mechanism (see Algorithm 2) iterates over $q_i \in \{0, 1, \dots, \lceil \log T \rceil\}$ and at every time step q_i , a required stage-service of $R' = R/2^i$ is applied to allocate sensing services (illustrated in Fig. 2). This means that R' services should be allocated before the end of this stage. Finally, the required services R should be allocated before the end of the deadline T . At the beginning of the mechanism, we introduce a small value ε as initial density threshold. We assume that the marginal value of user i ($i \notin S$) is $V_i(S) = V(S \cup \{i\})$, where S is selected users' set. In the sequel, as long as the arrival user's marginal density $\frac{V_i(S)}{b_i}$ is not less than the current threshold density value ρ^* and the budget has not been exhausted, the mechanism allocates service to it. Meanwhile, we give user i a payment $V_i(S)/\rho^*$, and add this user to the set of selected users S .

In the computation of the density threshold for the mechanism, we first find the maximal density for fulfilling $\delta R'$ services from the sample set S' . Then the process is repeated by using a simple greedy manner until all of $\delta R'$ services are allocated. The greedy manner sorts users according to their density, preferentially allocates services to users with higher density. Here, we set δ to blow up the required stage services so that the constant blowup services can be allocated at the next stage. Furthermore, we compute the total payment for fulfilling the constant blowup services. Furthermore, the algorithm calls the following the budget feasible mechanism for submodular function and then sets the density

Algorithm 2 SOS// Service-constraint Online incentive mechanism under a Sequential arrival model

Input: Service constraint R , sensing task deadlines T

```

1:  $(t, T', R', S', \rho^*, S) \leftarrow (1, \frac{T}{2^{\lceil \log_2 T \rceil}}, \frac{R}{2^{\lceil \log_2 T \rceil}}, \emptyset, \varepsilon, \emptyset)$ ;
2: for  $t \leq T$  do
3:   if there is a user  $i$  arriving at time step  $t$  then
4:     if  $b_i \leq V_i(S)/\rho^*$  and  $V(S) < R'$  then
5:        $p_i \leftarrow V_i(S)/\rho^*$ ,  $S = S \cup \{i\}$ ;
6:     else
7:        $p_i \leftarrow 0$ ;
8:     end if
9:      $S' \leftarrow S' \cup \{i\}$ ;
10:  end if
11:  if  $t = \lfloor T' \rfloor$  then
12:    Calculate  $\rho^* \leftarrow \text{getDensityThreshold}(R', S')$ ;
13:    set  $R' \leftarrow 2R'$ ,  $T' \leftarrow 2T'$ ;
14:  end if
15:   $t \leftarrow t + 1$ ;
16: end for
```

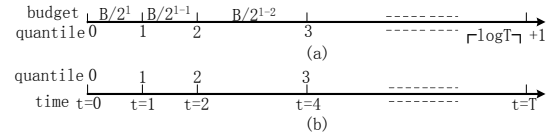


Fig. 2 Illustration of a multi-stage sample process with deadlines T . (a) Budget constraints over quantiles; (b) Quantiles over time.

threshold to be ρ/ν . ν is introduced to guarantee enough users selected and avoid the waste of payment.

The above budget feasible mechanism for submodular function is an offline mechanism proposed in [16]. It adopts a proportional share allocation rule [16] to compute the density threshold from the sample set S' and the budget B' . First of all, users are sorted according to their increasing marginal densities. In this sorting the $(i+1)$ -th user is the user j such that $V_j(S_i)/b_j$ is maximized over $S' \setminus S_i$, where $S_i = \{1, 2, \dots, i\}$ and $S_0 = \emptyset$. Considering the submodularity of V , this sorting implies that $\frac{V_1(S_0)}{b_1} \geq \frac{V_2(S_1)}{b_2} \geq \dots \geq \frac{V_{|S'|}(S_{|S'|-1})}{b_{|S'|}}$.

Then, the computation process adopts a greedy strategy. That is, according to increasing marginal contributions relative to their bids from the sample set to find the largest k satisfying $b_{k^*} \leq \frac{R' V_k(S_{k-1})}{V(S_k)}$. Furthermore, we can obtain the payment threshold estimated based on every sample set S' with the privacy profile of users and the allocated stage-budget R' . Finally, we set the density threshold to be $\frac{V(S_k)}{\delta R'}$. The detailed computation of the threshold density is illustrated in Algorithm 3 and Fig. 2.

We now prove that our mechanism satisfies the desirable properties as follows:

Lemma 1 *The SOS mechanism is incentive compatible or truthful.*

Proof To see that bid-independent auctions are truthful, here consider a user i with cost of c_i that arrives at some

Algorithm 3 getDensityThreshold**Input:** Sample user set S' , the stage-service R' .**Output:** The threshold density ρ .

```

1: Initialize:  $\mathcal{J}' \leftarrow \emptyset$ ;  $i \leftarrow \arg \max_{j \in S'} \frac{V_j(\mathcal{J})}{b_j}$ ;
2: while  $V(\mathcal{J}) < \delta R'$  do
3:    $\mathcal{J} \leftarrow \mathcal{J} \cup \{i\}$ ;
4:   Compute  $i \leftarrow \arg \max_{j \in S' \setminus \mathcal{J}} \frac{V_j(\mathcal{J})}{b_j}$ ;
5: end while
6:  $B \leftarrow \sum_{j \in \mathcal{J}} b_j$ ;
7:  $\rho \leftarrow \text{getFeasibleDensity}(B', S')$ ;
8: return  $\rho/\nu$ ;

```

Algorithm 4 getFeasibleDensity [16]**Input:** Sample user set S' , the budget B' .**Output:** The threshold density ρ .

```

1: Initialize:  $\mathcal{J}' \leftarrow \emptyset$ ;  $i \leftarrow \arg \max_{j \in S'} \frac{V_j(\mathcal{J})}{b_j}$ ;
2: while  $b_i \leq \frac{B' V_i(\mathcal{J})}{V(\mathcal{J} \cup \{i\})}$  and  $V(\mathcal{J}) \leq B'$  do
3:    $\mathcal{J} \leftarrow \mathcal{J} \cup \{i\}$ ;
4:   Compute  $i \leftarrow \arg \max_{j \in S' \setminus \mathcal{J}} \frac{V_j(\mathcal{J})}{b_j}$ ;
5: end while
6:  $\rho \leftarrow V(\mathcal{J})/B'$ ;
7: return  $\rho$ ;

```

stage for which the threshold density was set to ρ^* . If by the time the user arrives there are no remaining required stage services, then the user's cost declaration will not affect the allocation of the mechanism and thus cannot improve his utility by submitting a false cost. Otherwise, assume there are remaining required stage services by the time the user arrives. In case $c_i \leq V_i(S)/\rho^*$, reporting any cost below $V_i(S)/\rho^*$ wouldn't make a difference in the user's allocation and payment and his utility for each assignment would be $V_i(S)/\rho^* - c_i \geq 0$. Declaring a cost above $V_i(S)/\rho^*$ would make the user lose the auction, and his utility would be 0. In case $c_i > V_i(S)/\rho^*$, declaring any cost above $V_i(S)/\rho^*$ would leave the user unallocated with utility 0. If the user declares a cost lower than $V_i(S)/\rho^*$ he will be allocated. In such a case, however, his utility will be negative. Thus the user's utility is always maximized by reporting his true cost: $b_i = c_i$. Putting these discussions together, the SOS mechanism satisfies bid-independence. According to Proposition 2.1 in [2], i.e., if and only if an online auction is bid-independent, it is truthful. Thus, Lemma 1 holds.

Lemma 2 *The SOS mechanism is service feasible.*

Proof At each stage $t \in \{0, 1, \dots, \lfloor \log_2 T \rfloor, \lfloor \log_2 T \rfloor + 1\}$, the mechanism uses a stage-service of $R' = \frac{2^{t-1}R}{2^{\lfloor \log_2 T \rfloor}}$. From the lines 4-5 of Algorithm 2, we can see that it is guaranteed that the current total allocated services does not exceed the stage-service R' . Specially, the service constraint of the last stage is R . Therefore, every stage is service feasible, and when the deadline T arrives, the total allocated services does not exceed R . It is possible

that the total required services can not be fulfilled. To the end, we compute the minimal cost for fulfilling a constant blowup of the required services by a frugal ratio δ (see Algorithm 3). As such, $R/2$ required services could be allocated at the last stage while the total payment is no more than the budget B . Thereby, the mechanism can guarantee that each stage uses minimal payments to achieving the required stage services by blowing up to $\delta R'$ until the total required services are fulfilled. Thus, Lemma 2 holds.

Lemma 3 *The SOS mechanism is computational efficient.*

Proof Since the mechanism runs online, we only need to focus on the computation complexity at each time step $t = \{1, 2, \dots, T\}$. Computing the marginal value of user i takes $O(\Gamma_i)$ time, which is at most $O(m)$. Thus, the running time of computing the allocation and payment of user i (lines 3-10 of Algorithm 2) is bounded by $O(m)$. Next, we analyze the complexity of computing the density threshold, namely Algorithm 3. Finding the user with maximum marginal density takes $O(m|S'|)$ time. Since there are m tasks and each selected user should contribute at least one new task, the number of winners is at most $\min\{m, |S'|\}$. Thus, the running time of lines 1-6 of Algorithm 3 is bounded by $O(m|S'| \min\{m, |S'|\})$. The running time of line 7 of Algorithm 3 is the same as of lines 1-6 of Algorithm 3. Thus, the computation complexity at each time step (lines 3-15) is bounded by $O(m|S'| \min\{m, |S'|\})$. At the last stage, the sample set S' has the maximum number of samples, being $n/2$ with high probability. Thus, the computation complexity at each time step is bounded by $O(mn \min\{m, n\})$. Thus, Lemma 3 holds.

Lemma 4 *The SOS mechanism is individually rational.*

Proof From the lines 4-7 of Algorithm 2, we can see that $p_i \geq b_i$ if $i \in S$, otherwise $p_i = 0$. Therefore, we have individual gain $u_i \geq 0$. Thus, Lemma 4 holds.

Lemma 5 *The SOS mechanism satisfies the consumer sovereignty.*

Proof Each stage is an accepting process as well as a sampling process ready for the next stage. As a result, users are not automatically rejected during the sampling process, and are allocated as long as their marginal densities are not less than the current threshold density, and the allocated stage services has not been exhausted. Thus, Lemma 5 holds.

If the stage services could be achieved at each stage, then R required services would be allocated finally. Since our SOS mechanism consists of multiple stages, and dynamically increases the stage services, it only needs to prove that $R/2$ required services could be allocated at the last stage while the total payment is no more than

the budget B . Thereby, the mechanism can guarantee that each stage uses minimal payments to achieving the required stage services by blowing up to $\delta R'$ until the total required services are fulfilled. The frugality ratio for achieving the required services would be δ , since at the last stage the budget B is the minimal cost for fulfilling the required stage services $\delta R' = \delta R/2$ according to Algorithm 3. The mechanism for minimizing payments is originated from the observations that the stage-service constraint at each stage can be changed into the budget constraint at the correspondent stage. If we show that at least $R/2$ required services could be allocated at the last stage under the budget constraint B , then it is equivalent to that $R/2$ required services could be allocated while the total payment is no more than B . This means that the frugality ratio for achieving the required services is δ .

Lemma 6 *The SOS mechanism satisfies $O(1)$ -competitive, i.e., constant frugal ratio. Specifically, under i.i.d. model, we can achieve the announced services from the platform when the frugal ratio $\delta = 8$. Under the secretary model, we can achieve the announced services from the platform when the frugal ratio $\delta = 24$.*

The detailed proof is given in Appendix A. From the above lemmas, the following theorem holds.

Theorem 2 *The SOS mechanism satisfies computational efficiency, individual rationality, service feasibility, truthfulness, consumer sovereignty, and constant frugality under a sequential arrival model.*

6 Performance Evaluation

To evaluate the performance of our service-constraint mechanisms, we implemented the OMS and SOS mechanisms, and compared them against the random mechanism, i.e., uses a simple greedy algorithm like Algorithm 3, which adopts a naive strategy for rewarding users based on an uninformed fixed bid threshold. The performance metrics include the frugal ratio, the running time, and the platform's value.

6.1 Simulation Setup

We set the deadline (T) to 1800s, and vary the required services (R) from 200 to 2000 with the increment of 200. Users arrive according to a Poisson process in time with arrival rate λ . We vary λ from 0.2 to 1 with the increment of 0.2. The sensing range of each sensor is set to 7 meters. The cost of each user is uniformly distributed over $[1, 10]$. The initial density threshold (ϵ) of Algorithm 1 and 4 is set to 1. Note that this threshold could be an empirical value for real applications. All the simulations were run on a PC with 1.7 GHz CPU and 8 GB memory. Each measurement is averaged over 100 instances.

All the simulations were run on a PC with 1.7 GHz CPU and 8 GB memory. Each measurement is averaged over 100 instances.

6.2 Evaluation Results

We first evaluate the frugal ratio's impact on the OMS and SOS mechanisms. Then when the frugal ratio is fixed, we evaluate their performances against the random mechanism.

Comparison on total payments: The total payments of all evaluated mechanisms increase with the value of required services. From Fig. 3, we can observe that the payments of the SOS mechanism ($\delta = 6$) is lower than optimal offline mechanism with $6R$ services. Note that, at most 4107 services can be completed by the OMS mechanism in our simulations due to the limit of the number of arrival users, and the 801 services can be completed by the the SOS mechanism ($\delta = 6$) and the platform's payment is 4657.5 when the value of required services is set as 800, while the 3603 services can be completed by OMS, and the platform's payment is 13940 under there are $6R = 4801$ required services. The payment of SOS mechanism is much lower than one of mechanism. This shows that the "realistic" frugality ratio is less than 6, which is consistent with our theoretical analysis in Lemma 6. Thus, as the required services increase, the mechanism SOS have lower payments than the OMS mechanism. However, as the value of the required services increases, the payments of the mechanism SOS are larger than the OMS mechanism. It is because there is a limit of the number of available users in the system. Additionally, Although Fig. 3 shows that random online mechanism has lower payments than our mechanisms, our mechanisms ensure that required services are completed when there are enough users to select. When the value of required services is equal to 1200, the services completed by the random online mechanism are lower than half of required services, i.e. 591.

Frugal ratio's impact: Fig. 4(a) shows that the density threshold of each stage decreases as the frugal ratio δ increases, thereby achieving much lower payments. The density threshold of each stage tends to a constant when the frugal ratio δ is larger than 8. Thus, the SOS mechanism learns a optimal density that achieves the minimal payments meanwhile fulfilling required services. The SOS mechanism can attain lower payment as the value of δ increases. When the payments of the SOS mechanism are lower than the payments, we call the value as the frugal ratio. Fig. 4(b) shows that the total payments of the platform converges towards a constant value with the increase of the frugal ratio δ .

Required service value's impact: Fig. 5(a) shows that completed services of random online mechanism are much lower that required services. However, the SOS mechanism ensures that required services are completed

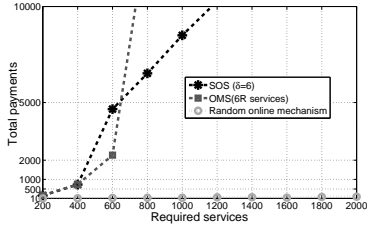


Fig. 3 Total payments versus required services.

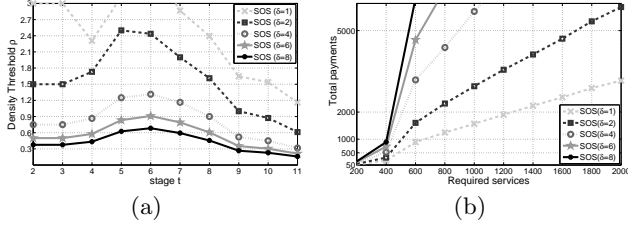


Fig. 4 (a) Density threshold ρ versus stage t under different frugal ratio δ when the value of required services is set as 1000; (b) The total payments versus required services under different frugal ratio δ .

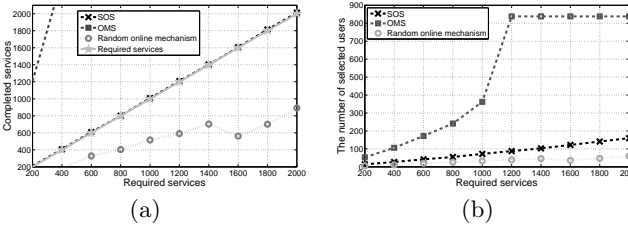


Fig. 5 (a) Completed services versus required services; (b) The number of selected users versus required services when $\delta = 6$.

when there are enough users to select. When the value of required services are more than the value 1000, SOS mechanism completes all required services while OMS mechanism does not complete 6R required services. It is because that there is the limitation of available users in simulation (here the number of available users is set as 838). However, the limit of the number of available users has no impact on the SOS mechanism, because in SOS mechanism only need to select 163 users to complete the required services 2000, as shown in Fig. 5(b).

7 Conclusions

In this paper, we have designed two incentive mechanisms to motivate smartphone users to participate in crowd sensing application with the service constraint, which is a new sensing paradigm allowing us to efficiently collect data for achieving required service quality. We first propose a OMS mechanism for the offline scenario. Furthermore, we design a SOS mechanism for a sequential arrival model, where users arrive one by one online. We also prove that the two mechanisms satisfy the above desirable properties.

References

1. Badanidiyuru A, Kleinberg R, Singer Y (2012) Learning on a budget: posted price mechanisms for online procurement. In: Proceedings of ACM EC, pp 128–145
2. Bar-Yossef Z, Hildrum K, Wu F (2002) Incentive-compatible online auctions for digital goods. In: Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms, Society for Industrial and Applied Mathematics, pp 964–970
3. Bateni M, Hajiaghayi M, Zadimoghaddam M (2010) Submodular secretary problem and extensions. In: Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, Springer, pp 39–52
4. Chen Y, Vaughan JW (2010) A new understanding of prediction markets via no-regret learning. In: Proceedings of the 11th ACM conference on Electronic commerce, pp 189–198
5. Dynkin EB (1963) The optimum choice of the instant for stopping a markov process. In: Proceedings of Soviet Math. Dokl, vol 4
6. Hajiaghayi MT, Kleinberg R, Parkes DC (2004) Adaptive limited-supply online auctions. In: Proceedings of the 5th ACM conference on Electronic commerce, ACM, pp 71–80
7. Ho CJ, Vaughan JW (2012) Online task assignment in crowdsourcing markets. In: Proceedings of AAAI
8. Horton JJ, Zeckhauser RJ (2010) Algorithmic wage negotiations: Applications to paid crowdsourcing. In: Proceedings of CrowdConf, vol 4
9. Kleinberg R (2005) A multiple-choice secretary algorithm with applications to online auctions. In: Proceedings of ACM SIAM, pp 630–631
10. Koukoulidis E, Peh LS, Martonosi MR (2011) Signaguru: leveraging mobile phones for collaborative traffic signal schedule advisory. In: Proceedings of the 9th international conference on Mobile systems, applications, and services, ACM, pp 127–140
11. Maisonneuve N, Stevens M, Niessen ME, Steels L (2009) Noisetube: Measuring and mapping noise pollution with mobile phones. In: Information Technologies in Environmental Engineering, Springer, pp 215–228
12. Mason W, Watts DJ (2010) Financial incentives and the performance of crowds. ACM SigKDD Explorations Newsletter 11(2):100–108
13. Mohan P, Padmanabhan VN, Ramjee R (2008) Nericell: using mobile smartphones for rich monitoring of road and traffic conditions. In: Proceedings of the 6th ACM conference on Embedded network sensor systems, ACM, pp 357–358
14. Rana RK, Chou CT, Kanhere SS, Bulusu N, Hu W (2010) Ear-phone: an end-to-end participatory urban noise mapping system. In: Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks, ACM, pp 105–116
15. Shaw AD, Horton JJ, Chen DL (2011) Designing incentives for inexpert human raters. In: Proceedings of the ACM 2011 conference on Computer supported cooperative work, pp 275–284
16. Singer Y (2010) Budget feasible mechanisms. In: Proceedings of IEEE FOCS, pp 765–774
17. Singer Y, Mittal M (2013) Pricing mechanisms for crowdsourcing markets. In: Proceedings of ACM WWW, pp 1157–1166
18. Singla A, Krause A (2013) Truthful incentives in crowdsourcing tasks using regret minimization mechanisms. In: Proceedings of ACM WWW, pp 1167–1177
19. Thiagarajan A, Ravindranath L, LaCurts K, Madden S, Balakrishnan H, Toledo S, Eriksson J (2009) Vtrack: accurate, energy-aware road traffic delay estimation using

- mobile phones. In: Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems, ACM, pp 85–98
20. Tran-Thanh L, Stein S, Rogers A, Jennings NR (2012) Efficient crowdsourcing of unknown experts using multi-armed bandits. In: Proceedings of European Conference on Artificial Intelligence, pp 768–773
 21. Yang D, Xue G, Fang X, Tang J (2012) Crowdsourcing to smartphones: incentive mechanism design for mobile phone sensing. In: Proceedings of ACM MobiCom, pp 173–184
 22. Zhao D, Li X, Ma H (2013) OMG: how much should I pay bob in truthful online mobile crowdsourced sensing? CoRR URL <http://arxiv.org/abs/1306.5677>

A

Proof of Lemma 6:

Let S^* be the set of users selected by the offline Algorithm 4 before the time T and the budget $2B$, the value of S^* is $V(S^*)$. The value density threshold of S^* is $\rho = V(S^*)/B$. S' is the sample set obtained at the time $T/2$. $S_1^* = S^* \cap S'$ and $S_2^* = S^* \cap \{\mathcal{U} \setminus S'\}$. S'_1 is the set of users selected from the sample set S' by Algorithm 4 before the time T and the budget B , and S'_2 is the set of users selected by Algorithm 2 at the last stage. Let $\rho'_1 = V(S'_1)/B$ be the density computed using Algorithm 3 over S' and $\rho^* = \rho'_1/\nu$ is the density threshold of the last stage. Assume that the value of each user is at most $\max_i V_i \leq V(S^*)/\omega$.

Proof In the proof, we consider that the mechanism is constant frugal from the two class model: I.I.D. and the Secretary Model

Under I.I.D. Model, since the costs and values of all users in \mathcal{U} are i.i.d., they can be selected in the set S^* with the same probability. Thus, we have $\mathbb{E}[|S'_1|] = \mathbb{E}[|S'_2|] = |S^*|/2$. Considering the submodularity of function $V(S)$, it can be derived that: $\mathbb{E}[V(S'_1)] \geq \mathbb{E}[V(S'_2)] \geq V(S^*)/2 = R/2$. Since $V(S'_1)$ is computed with the stage-budget $B/2$, it can be derived that: $\mathbb{E}[V(S'_1)] \geq \mathbb{E}[V(S_1^*)] \geq V(S^*)/2 = R/2$ and $\mathbb{E}[\rho'_1] \geq \rho$, where the first inequality follows from the fact that $V(S'_1)$ is the optimal solution computed by Algorithm 4. Therefore, we only need to prove that the ratio of $\mathbb{E}[V(S'_2)]$ to $\mathbb{E}[V(S'_1)]$ is at least a constant, then the SOS mechanism have a constant frugal ratio. Only two cases can exist according to the total payment to the selected users at the last stage.

According to Lemma 7 in [22], we have $1/2 - (\frac{\nu}{1-2\alpha} - 1)/\omega - 1/\nu = 2\alpha/\nu$. Thus, when ω is sufficiently large (at least 12), we can obtain a constant ratio of $\mathbb{E}[V(S'_2)]$ to $\mathbb{E}[V(S'_1)]$. More importantly, the optimal ratio increases to $1/4$ (i.e., $2\alpha/\nu \rightarrow 1/4$) as ω increases.

From Lemma 9 in [22], we have $\mathbb{E}[V(S'_1)] \geq \frac{\delta R}{4}$. Furthermore, $\mathbb{E}[V(S'_2)] \geq \frac{2\alpha}{\nu} \mathbb{E}[V(S'_1)] \geq \frac{2\alpha}{\nu} \cdot \frac{\delta R}{4}$. According to the previous discussions, to achieve the required services, the inequality $\mathbb{E}[V(S'_2)] \geq R/2$ holds by setting $\frac{\alpha\delta R}{2\nu} \geq \frac{R}{2}$. As such, we have $\delta \geq 2 \cdot \nu/2\alpha \geq 2 \times 4 = 8$. Thus, we can set the frugal ratio $\delta = 8$ to achieve the required services.

Under the Secretary Model, let S^* be the set of users selected by the offline Algorithm 4 before the time T and the budget B other than the budget $2B$ in the i.i.d. model. According to Lemma 15 in [3], for sufficiently large ω , the random variable $|V(S_1^*) - V(S_2^*)|$ is bounded by $V(S^*)/2$ with

a constant probability. Because of the submodularity of V , we have $V(S_1^*) + V(S_2^*) \geq V(S^*)$. Thus, we easily obtain the result: For sufficiently large ω , both $V(S_1^*)$ and $V(S_2^*)$ are at least $V(S^*)/4$ with a constant probability. Putting the result and Lemma 9 in [22] together, we have $\geq V(S_1)/2 \geq V(S^*)/8$. Only two cases can exist according to the total payment to the selected users at the last stage.

According to Lemma 10 in [22], we have $1/4 - (\frac{8\nu}{1-2\alpha} - 1)/\omega - 2/\nu = 2\alpha/\nu$. Thus, when ω is sufficiently large (at least 12), we can obtain a constant ratio of $V(S'_2)$ to $V(S'_1)$. More importantly, the optimal ratio increases to $1/12$ (i.e., $2\alpha/\nu \rightarrow 1/12$) as ω increases.

In terms of Lemma 9 in [22], we have $V(S'_1) \geq \frac{\delta R}{4}$. Furthermore, $V(S'_2) \geq \frac{2\alpha}{\nu} V(S'_1) \geq \frac{2\alpha}{\nu} \cdot \frac{\delta R}{4}$. According to the previous discussions, to achieve the required services, the inequality $V(S'_2) \geq R/2$ holds by setting $\frac{\alpha\delta R}{2\nu} \geq \frac{R}{2}$. As such, we have $\delta \geq 2 \cdot \nu/2\alpha \geq 2 \times 12 = 24$. Thus, we can set the frugal ratio $\delta = 24$ to achieve the required services.

Thus, the Lemma 6 holds.

